

MODIFIED FILON-CLENSHAW-CURTIS RULES FOR OSCILLATORY INTEGRALS WITH A NONLINEAR OSCILLATOR

HASSAN MAJIDIAN

ABSTRACT. Filon-Clelland-Curtis rules are among rapid and accurate quadrature rules for computing highly oscillatory integrals. In the implementation of the Filon-Clelland-Curtis rules in the case when the oscillator function is not linear, its inverse should be evaluated at some points. In this paper, we solve this problem by introducing an approach based on the interpolation, which leads to a class of modifications of the original Filon-Clelland-Curtis rules. In the absence of stationary points, two kinds of modified Filon-Clelland-Curtis rules are introduced. For each kind, an error estimate is given theoretically, and then illustrated by some numerical experiments. Also, some numerical experiments are carried out for a comparison of the accuracy and the efficiency of the two rules. In the presence of stationary points, the idea is applied to the composite Filon-Clelland-Curtis rules on graded meshes. An error estimate is given theoretically, and then illustrated by some numerical experiments.

Keyword. Filon-Clelland-Curtis rule; Highly oscillatory integral; Nonlinear oscillator; Barycentric interpolation; Graded mesh.

MSC 65D30; 65T40

1. INTRODUCTION

Consider integrals of the form

$$I_k^{[a,b]}(f, g) := \int_a^b f(x) \exp(ikg(x)) dx, \quad (1)$$

where $f \in L^1[a, b]$, $k > 0$, and $g \in C^m[a, b]$ for some $m \geq 1$. If g does not oscillate rapidly in $[a, b]$, the integrand in (1) oscillates violently for larger values of k . This class of integrals contains a large portion of highly oscillatory integrals, appearing in many areas of science and engineering, e.g. Fourier series and transforms, special functions, high-frequency acoustic scattering, etc (see, e.g. [18, 5] and references therein). Because of their wide applications, computing highly oscillatory integrals of the form (1) has been the subject of many researches in the last two decades (see the references at the end of this article).

Filon-types methods are among most efficient ones for computing (1) with a long history. While high accurate methods based on steepest descents (see, e.g. [13, 7, 12]) needs some manual calculations regarding the steepest decent paths in the complex plane, the Filon-type methods does not deals with the complex calculus and can be performed automatically by computers provided that the moments can be computed to the desired accuracy. The idea, in general, is to replace the amplitude function $f(x)$ by an interpolating polynomial $p_n(x)$ and consider $I_k^{[a,b]}(p_n, g)$ as an approximation of $I_k^{[a,b]}(f, g)$. This idea originated from [10], where Louis Napoleon George Filon (1875–1937) applied it to the Fourier integral

$$\int_a^b f(x) \sin \omega x dx.$$

After the work of Filon, many papers on his method appeared, but the 2005 paper [14] proved to be a milestone in the history of researches on the Filon-type methods. In this work, the asymptotic expansion of $I_k^{[a,b]}(f, g)$ for increasing k has been studied, and a generalization of the Filon's method has been provided that converges as negative powers of k . The paper has been the motivation of many later works.

The implementation of a Filon-type method rests on the ability to compute the moments $I_k^{[a,b]}(x^m, g)$, as mentioned above. For the linear oscillator $g(x) = x$, the moments can be computed by the following identity:

$$I_k^{[a,b]}(x^m, x) = \frac{1}{(-ik)^{m+1}} [\Gamma(1+m, -ika) - \Gamma(1+m, -ikb)], \quad (2)$$

where Γ is the incomplete Gamma function [1]. For a more complex oscillator g , S. Olver [17] considered a g -dependent basis instead of the usual polynomial basis x^j . Then the moments can be written in closed form, again in term of incomplete Gamma functions. Another approach, proposed in [28], is based on the transformation $\tau = g(x)$. The method enables one to compute the moments by the identity (2) when g has no stationary points in $[a, b]$; otherwise if g has a single stationary point, similar identities for $I_k^{[0,\tau]}(x^m, x^j)$ are employed (cf. [1, 14]).

Filon-Clenshaw-Curtis (FCC) rules, beside other advantages, enable one to compute the moments efficiently and rapidly without needing to deal with (incomplete) Gamma functions, whose evaluation needs a considerable number of flops. The N -point FCC rule can be described as follows. Consider the oscillatory integral

$$I_k(f) := \int_{-1}^1 f(x) \exp(ikx) dx. \quad (3)$$

In the N -point FCC rule, the amplitude function f is interpolated at Clenshaw-Curtis points $t_{j,N} := \cos(j\pi/N)$, $j = 0, \dots, N$ by the polynomial

$$Q_N f(s) := \sum_{n=0}^N {}'' \alpha_{n,N}(f) T_n(s), \quad (4)$$

where $T_n(s) = \cos(n \arccos(s))$ are the Chebyshev polynomials of the first kind, and $\sum {}''$ means that the first and the last terms of the sum are to be halved. By discrete orthogonality of the cosine function, the coefficients $\alpha_{n,N}(f)$ can be written as

$$\alpha_{n,N}(f) = \frac{2}{N} \sum_{j=0}^N {}'' \cos(jn\pi/N) f(t_{j,N}), \quad n = 0, \dots, N. \quad (5)$$

If f in (3) is replaced by $Q_N f$, then the N -point FCC rule is obtained as

$$I_{k,N}(f) = \sum_{n=0}^N {}'' \alpha_{n,N}(f) \omega_n(k), \quad (6)$$

where the weights (modified moments) $\omega_n(k) = I_k(T_n)$, $n \geq 0$, can be computed recursively thanks to the three-term recurrence relation for the Chebyshev polynomials T_n . As a generalization of the original Clenshaw-Curtis rules [6], this idea has been developed gradually by employing the results of many earlier works, mainly the Piessens' contributions [23, 20, 21, 22, 19] and the Sloan's works [26, 27]. By a simple affine change of variables, the FCC rules (6) can also be applied to

$$I_k^{[a,b]}(f) := \int_a^b f(x) \exp(ikx) dx. \quad (7)$$

The recent papers of Domínguez et al. [9, 8] contain important developments and results on the FCC rules. In [9], beside an error bound in term of both k and N , a

two-phase algorithm with the complexity of $\mathcal{O}(N \log N)$ was proposed for computing the weights, and also the stability of the algorithm was proved rigorously. The algorithm can be employed for computing the general integral (1) in the following way (cf. [8]): If g has no stationary points in $[a, b]$, then the change of variable $\tau = g(x)$ reduces the integral to $I_k^{[g(a), g(b)]}(F)$ with $F = (f \circ g^{-1})|(g' \circ g^{-1})|^{-1}$; otherwise, if g has some stationary points in $[a, b]$, similar change of variables reduce the integral to a finite number of integrals of the form (7), such that each integral has a single singularity at one of the endpoints and can be computed efficiently by the the method proposed in [8]: The integration interval is partitioned by a graded mesh with a large enough grading exponent; then on each panel (except the first one that contains the singular point) the N -point FCC rule is applied. Therefore, the method for computing the general integral (1) consists of several applications of the algorithm proposed in [9] for computing the regular integral (3). In other word, efficient performance of the two-phase algorithm of [9] is essential for computing the general integral (1).

Statement of the problem (or the motivation to this paper). In the implementation of the algorithm proposed in [8], computing $g^{-1}(x)$ is required at several points. Expressing $g^{-1}(x)$ in a closed form, however, is not always possible. For example, consider the function $g(x) = x - \sin(x)$. Thus, we should resort to Newton iterations, but it is well-known that their convergence is not guaranteed.

In this paper, we follow the idea of [28] and propose an efficient method for computing (1), that is based on the FCC rules, but without needing to compute $g^{-1}(x)$ at any point. Our main motivation to this work, lies in the automatization purposes indeed. While the algorithm proposed in [8] requires the inverse and the first derivative of g as inputs, the proposed algorithms in this papers require neither of them.

The plan of this paper is as follows. Section 2, contains the main idea of the method. There, we describe a class of quadrature rules for computing (1) in the absence of stationary points ($g(x) \neq 0$, for any $x \in [a, b]$). The rules can be considered as modifications of the FCC rules of [8], which release us from computing $g^{-1}(x)$ by imposing some interpolation process. Thus, an interpolating error is added to the total error. In order to decrease the interpolating error, two efficient interpolation methods are considered, base on which two kinds of modified FCC rules are proposed. The next two sections deal with these kinds of the quadrature rules, their error estimates, and some numerical experiments. Section 5 includes a comparison of the two kinds of the rules and their efficiency. In Section 6, the case when g has some stationary points in $[a, b]$ is studies; we show how the modified FCC rules can be applied in this case. An error estimate is provided, and some numerical results are given. Finally, we bring a conclusion.

Some general remarks. Throughout the paper, C and C' stand for generic constants independent of k , and their values may differ from place to place. Their dependency to other parameters (if exist) is declared in each section. Note also that the space of m times continuously differentiable functions is denoted by the common notation $C^m[a, b]$. Also, the reference values of the integrals in all of the numerical experiments, reported in this paper, have been computed by Mathematica to at least 10 digits higher than the machine precision.

2. THE GENERAL IDEA

Consider the integral (1), and assume that g has no stationary points in $[a, b]$. Then we can assume that $g'(x) > 0$, $x \in [a, b]$, without loss of generality. By this assumption, the integral is changed into an integral with the linear oscillator.

Indeed, by the change of variable $\tau = g(x)$,

$$\begin{aligned} I_k^{[a,b]}(f, g) &:= \int_{g(a)}^{g(b)} F(\tau) \exp(ik\tau) d\tau \\ &= l \exp(ikc) I_{\tilde{k}}(\tilde{F}) \\ &= l \exp(ikc) \int_{-1}^1 \tilde{F}(\tau) \exp(i\tilde{k}\tau) d\tau, \end{aligned} \quad (8)$$

where

$$F = (f \circ g^{-1}) (g' \circ g^{-1})^{-1}, \quad (9)$$

$$c = \frac{g(b) + g(a)}{2}, \quad l = \frac{g(b) - g(a)}{2}, \quad (10)$$

$$\tilde{k} = lk, \quad \tilde{F}(\tau) = F(c + l\tau). \quad (11)$$

The efficient and robust algorithm of [9], that is based on the FCC rules, can now be applied to (8). However, evaluating $g^{-1}(x)$ for some x is required in this process, and this is not of our favorite. For example, it may be impossible to express $g^{-1}(x)$ in a closed form. Then we should resort to Newton iterations, while their convergence is not guaranteed.

In the following, we develop a modification of the algorithm in such a way that the need for computing $g^{-1}(x)$ is released. The idea can then be employed in the presence of stationary points by means of the composite rules on graded meshes, as developed in [8]. This idea is discussed in Section 6.

2.1. The modified Filon-Clenshaw-Curtis (MFCC) rule. Employing the N -point FCC rule (6) for computing $I_{\tilde{k}}(\tilde{F})$ (that is essential in (8)) necessitates evaluation of \tilde{F} at the Clenshaw-Curtis points. We describe a method for accurate approximation of \tilde{F} at the Clenshaw-Curtis points, in which g^{-1} is not needed to be evaluated at any point.

Let N be a positive integer and consider $N+1$ Clenshaw-Curtis points $t_{0,N}, \dots, t_{N,N}$. Let $\ell_1 : [-1, 1] \rightarrow [a, b]$ and $\ell_2 : [-1, 1] \rightarrow [g(a), g(b)]$ be the onto lines. By definition, $\tilde{F} = F \circ \ell_2$.

For a large integer $N' > 1$, consider a set of points $-1 = u_0 < \dots < u_{N'} = 1$ and define

$$d_j := \ell_2^{-1}(g(\ell_1(u_j))), \quad j = 0, \dots, N'. \quad (12)$$

Then $-1 = d_0 < \dots < d_{N'} = 1$. One can compute $\tilde{F}(d_j)$ without needing to evaluate the inverse of g at any point. Indeed,

$$\tilde{F}(d_j) = F(\ell_2(d_j)) = \sigma_1[f, g](\ell_1(u_j)), \quad (13)$$

where

$$\sigma_1[f, g] := \frac{f}{g'}. \quad (14)$$

The derivative of $g(x)$ can also be computed accurately and stably by the complex step derivative approximation (see, e.g. [16]):

$$f'(x) \approx \frac{\Im f(x + ih)}{h}. \quad (15)$$

The formula (15) is of order 2, and it is numerically stable. Thus, for evaluating $\tilde{F}(d_j)$ by (13), neither g^{-1} nor g' is required to be known in closed forms.

Now we can approximate $\tilde{F}(t_{j,N})$ by interpolation of \tilde{F} at the nodes $d_0, \dots, d_{N'}$. Denote by $\tilde{Q}_{N'}$ the interpolating projection corresponding to some method of interpolation at the points $-1 = d_0 < \dots < d_{N'} = 1$. Then

$$\tilde{Q}_{N'} \tilde{F}(t_{j,N}) \approx \tilde{F}(t_{j,N}), \quad j = 0, \dots, N.$$

Now the MFCC rule is defined by

$$I_{\tilde{k},N}(\tilde{Q}_{N'} \tilde{F}) = \sum_{n=0}^N {}'' \alpha_{n,N}(\tilde{Q}_{N'} \tilde{F}) \omega_n(\tilde{k}), \quad (16)$$

where the coefficients are expressed by (5), i.e.

$$\alpha_{n,N}(\tilde{Q}_{N'} \tilde{F}) = \frac{2}{N} \sum_{j=0}^N {}'' \cos(jn\pi/N) \tilde{Q}_{N'} \tilde{F}(t_{j,N}), \quad n = 0, \dots, N. \quad (17)$$

Also, the weights $\omega_n(\tilde{k})$ are recursively computed by the following algorithm (see [9]):

$$\begin{aligned} \rho_1(\tilde{k}) &:= \gamma_0(\tilde{k}), \\ \rho_1(\tilde{k}) &:= 2\gamma_1(\tilde{k}) - \frac{2}{i\tilde{k}} \gamma_0(\tilde{k}), \\ \rho_{n+1}(\tilde{k}) &:= 2\gamma_n(\tilde{k}) - \frac{2n}{i\tilde{k}} \rho_n(\tilde{k}) + \rho_{n-1}(\tilde{k}), \quad n = 2, \dots, \min\{N, \tilde{k}\} - 1, \\ \omega_0(\tilde{k}) &:= \rho_1(\tilde{k}), \quad \omega_n(\tilde{k}) := \gamma_n(\tilde{k}) - \frac{n}{i\tilde{k}} \rho_n(\tilde{k}), \quad n = 1, 2, \dots, \min\{N, \tilde{k}\}, \end{aligned}$$

where

$$\gamma_n(\tilde{k}) = \begin{cases} \frac{2 \sin \tilde{k}}{\tilde{k}}, & \text{for even } n; \\ \frac{2 \cos \tilde{k}}{i\tilde{k}}, & \text{for even } n. \end{cases}$$

In summary, what we have done is to approximate the integral (1) through the following process:

$$I_k^{[a,b]}(f, g) = l \exp(ikc) I_{\tilde{k}}(\tilde{F}) \approx l \exp(ikc) I_{\tilde{k},N}(\tilde{F}) \approx l \exp(ikc) I_{\tilde{k},N}(\tilde{Q}_{N'} \tilde{F}).$$

2.2. Error analysis. The error of the MFCC rule is actually affected by the error of the involved interpolation method. In this subsection, we study how the interpolation error affects the total error of the MFCC rule. The results are then employed to choose efficient interpolation methods, such that the total error of the MFCC rule decays rapidly.

Consider the weighted seminorm

$$\|\varphi\|_{H_w^m[a,b]} := \left\{ \int_a^b \frac{|\varphi^{(m)}(x)|^2}{\sqrt{(b-x)(x-a)}} dx \right\}^{1/2}, \quad (18)$$

as introduced in [8]. Clearly, the Hilbert space $H_w^m[a,b]$, induced by $\|\cdot\|_{H_w^m[a,b]}$, contains $C^m[a,b]$.

Theorem 2.1. *Assume that f and g are so smooth that $F \in H_w^m[g(a), g(b)]$. Let $r \in [0, 2]$ and $0 \leq m \leq N+1$. Then the absolute error of the MFCC is bounded by*

$$\begin{aligned} \left| I_k^{[a,b]}(f, g) - l \exp(ikc) I_{\tilde{k},N}(\tilde{Q}_{N'} \tilde{F}) \right| &\leq C k^{-r} h^{m+1-r} N^{-m+\rho(r)} \\ &\quad + C' k^{-1} \sqrt{N} \sup_{x \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(x) \right|, \quad (19) \end{aligned}$$

where $h := b - a$ is the length of the integration interval,

$$\rho(r) = \begin{cases} r, & 0 \leq r \leq 1, \\ (5r - 3)/2, & 1 \leq r \leq 2, \end{cases}$$

and C, C' are constants independent of N .

Proof. The total error is bounded by

$$\begin{aligned} \left| I_k^{[a,b]}(f, g) - l \exp(ikc) I_{\tilde{k}, N}(\tilde{Q}_{N'} \tilde{F}) \right| &\leq \left| I_k^{[a,b]}(f, g) - I_{k, N}^{[g(a), g(b)]}(F) \right| \\ &\quad + l \left| I_{\tilde{k}, N}(\tilde{F} - \tilde{Q}_{N'} \tilde{F}) \right|. \end{aligned} \quad (20)$$

By assumption $F \in H_w^m[g(a), g(b)]$. According to Theorem 2.6 of [8],

$$\left| I_k^{[a,b]}(f, g) - I_{k, N}^{[g(a), g(b)]}(F) \right| \leq C \sigma_{m, N} k^{-r} l^{m+1-r} N^{-m+\rho(r)}, \quad (21)$$

for some $\sigma_{m, N} > 0$. Also according to Remark 2.4 of [8], $\sigma_{m, N} \rightarrow 1$ as $N \rightarrow \infty$. Hence, $C \sigma_{m, N}$ is bounded by a constant independent of N , and then by the mean value theorem on g ,

$$\begin{aligned} \left| I_k^{[a,b]}(f, g) - I_{k, N}^{[g(a), g(b)]}(F) \right| &\leq C k^{-r} l^{m+1-r} N^{-m+\rho(r)} \\ &\leq C k^{-r} h^{m+1-r} N^{-m+\rho(r)}. \end{aligned} \quad (22)$$

On the other hand,

$$\left| I_{\tilde{k}, N}(\tilde{F} - \tilde{Q}_{N'} \tilde{F}) \right| = |\boldsymbol{\omega}_N^T \mathbf{C}_N \boldsymbol{\delta}_N|, \quad (23)$$

where \mathbf{C}_N is the $(N+1) \times (N+1)$ matrix with the entries

$$\mathbf{C}_{n, j} = (2/N) \cos(jn\pi/N), \quad n, j = 0, \dots, N,$$

and $\boldsymbol{\omega}_N, \boldsymbol{\delta}_N$ are column vectors defined by

$$\boldsymbol{\omega}_N = [\omega_0(\tilde{k})/2, \omega_1(\tilde{k}), \dots, \omega_{N-1}(\tilde{k}), \omega_N(\tilde{k})/2]^T,$$

$$\boldsymbol{\delta}_N = \left[\frac{(\tilde{F} - \tilde{Q}_{N'} \tilde{F})(t_{0, N})}{2}, (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(t_{1, N}), \dots, (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(t_{N-1, N}), \frac{(\tilde{F} - \tilde{Q}_{N'} \tilde{F})(t_{N, N})}{2} \right]^T.$$

(see Remark 2.1 of [9]).

By Cauchy-Schwarz inequality, Eq. (23) yields

$$\left| I_{\tilde{k}, N}(\tilde{F} - \tilde{Q}_{N'} \tilde{F}) \right| \leq \|\boldsymbol{\omega}_N\|_2 \|\mathbf{C}_N \boldsymbol{\delta}_N\|_2. \quad (24)$$

The well-know asymptotic expansion of highly oscillatory integrals in the absence of stationary points (see, e.g. [14]) yields

$$|I_{\tilde{k}}(T_n)| = \mathcal{O}(\tilde{k}^{-1}),$$

and then $\|\boldsymbol{\omega}_N\|_2 \leq C' l^{-1} k^{-1} \sqrt{N}$.

On the other hand,

$$\begin{aligned}\|\mathbb{C}_N \delta_N\|_2 &= \frac{2}{N} \left\{ \sum_{j=0}^N \left(\sum_{n=0}^N {}'' (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(t_{n,N}) \cos \frac{jn\pi}{N} \right)^2 \right\}^{\frac{1}{2}} \\ &\leq \frac{2}{N} \sup_{x \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(x) \right| \left\{ \sum_{j=0}^N \left(\sum_{n=0}^N {}'' \cos \frac{jn\pi}{N} \right)^2 \right\}^{\frac{1}{2}} \\ &\leq 2 \sup_{x \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(x) \right|,\end{aligned}$$

because

$$\sum_{n=0}^N {}'' \cos \frac{mn\pi}{N} = \begin{cases} N, & m = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Thus,

$$\left| I_{k,N}(\tilde{F} - \tilde{Q}_{N'} \tilde{F}) \right| \leq C' l^{-1} k^{-1} \sqrt{N} \sup_{x \in [-1,1]} \left| (\tilde{F} - \tilde{Q}_{N'} \tilde{F})(x) \right|, \quad (25)$$

and then the result follows by (20). \square

As it is seen by (19), the total error of the MFCC rule is bounded by the sum of two terms: The first term is nothing but the error of the FCC rule, studied in details in [9, 8]; it decays rapidly as k and/or N grows. The second term corresponds to the interpolation process, that may be quite large due to the Runge effect. In the sequel, we present two strategies in order to treat the latter problem and obtain error bounds which decay rapidly by increasing k and/or N .

3. NO STATIONARY POINTS: METHOD I

One idea for treating the Runge phenomenon lies in splines. In this section, we present a method of integration based on *composite* MFCC rules on the uniform meshes. Without loss of generality, we consider the integral (1) over the standard interval $[0, 1]$. Throughout this section, assume that $g'(x) > 0$, $x \in [0, 1]$.

3.1. Algorithm I. Take a large integer M and divide the integration interval $[0, 1]$ into M subintervals of equal lengths h :

$$x_n = nh, \quad n = 0, \dots, M. \quad (26)$$

Then,

$$g(0) = g(x_0) < \dots < g(x_M) = g(1).$$

Assume that $f \in C^{N+1}[0, 1]$ and $g \in C^{N+2}[0, 1]$ for some integer $N \geq 1$. Then it is easy to see that F , as defined by (9), lies in $C^{N+1}[g(0), g(1)]$. In each panel, we apply the N -point MFCC rule (as described in §2.1) with $N' = N$, $u_j = t_{j,N}$, and \tilde{Q}_N being the Lagrange interpolating projection at the points $-1 = d_0 < \dots < d_N = 1$. Then the accuracy of the rule may be studied as follows.

Take an arbitrary subinterval $[x_{n-1}, x_n]$, and for the sake of simplicity of the notations set $[a, b] := [x_{n-1}, x_n]$. It is easy to see that

$$d_{j,N} - d_{j-1,N} = \frac{b-a}{g(b)-g(a)} g'(\eta_j) (t_{j,N} - t_{j-1,N}), \quad (27)$$

where $\eta_j \in [a, b]$. Thus, the distribution of the points $d_{j,N}$ is the same as that of the points $t_{j,N}$ for any interval $[a, b]$ with sufficiently small length h . This is because

$$\frac{b-a}{g(b)-g(a)} g'(\eta_j) \simeq 1,$$

by the mean value theorem. In this case,

$$\sup_{x \in [-1, 1]} \left| (\tilde{F} - \tilde{Q}_N \tilde{F})(x) \right| \leq CK^{-N},$$

for some constants C and $K > 1$, independent of N (see, e.g. [3]). In the most pessimistic case, when the Gaussian distribution of the nodes is ignored, the interpolating error is bounded by

$$\sup_{x \in [-1, 1]} \left| (\tilde{F} - \tilde{Q}_N \tilde{F})(x) \right| \leq (2l)^{N+1} \max_{x \in [g(a), g(b)]} \left| F^{(N+1)}(x) \right|.$$

This is because $F \in C^{N+1}[g(a), g(b)]$ and then $\tilde{F} \in C^{N+1}[-1, 1]$. Now, by the mean value theorem on g ,

$$\sup_{x \in [-1, 1]} \left| (\tilde{F} - \tilde{Q}_N \tilde{F})(x) \right| \leq C' h^{N+1}, \quad (28)$$

for N large enough. Here, C' is a generic constant independent of M , but dependent of N .

Since $\tilde{F} \in C^{N+1}[-1, 1] \subseteq H_w^{N+1}[-1, 1]$, Theorem 2.1 with $m = N + 1$ implies that the total error in the arbitrary panel $[a, b]$ is estimated by

$$\begin{aligned} \left| I_k^{[a, b]}(f, g) - l \exp(ikc) I_{k, N}(\tilde{Q}_N \tilde{F}) \right| &\leq C k^{-r} h^{N+2-r} N^{-N-1+\rho(r)} \\ &\quad + C' k^{-1} \sqrt{N} h^{N+1}, \end{aligned} \quad (29)$$

where C is a generic constant independent of M and N .

If one takes sum over all the subintervals $[x_{n-1}, x_n]$ and take into account that $F \in C^{N+1}[g(0), g(1)]$, an error bound for Algorithm I is obtained:

$$C k^{-r} h^{N+1-r} N^{-N-1+\rho(r)} + C' k^{-1} \sqrt{N} h^N. \quad (30)$$

A few words about the complexity of the algorithm is necessary. The N -point FCC rule can be stably implemented with $\mathcal{O}(N \ln N)$ flops [9]. The Lagrange interpolation can also be performed stably by an improved formula, the so-called the “first form of the barycentric interpolation formula” (see [25, 3, 11]); the cost of evaluation of the formula at any point is $\mathcal{O}(N^2)$, so evaluation of all the values $\tilde{Q}_N \tilde{F}(t_{j, N})$, for $j = 0, \dots, N$, requires $\mathcal{O}(N^3)$ flops. Since this cost should be paid for each panel, and there exist M panels indeed, Algorithm I requires $\mathcal{O}(MN(N^2 + \ln N))$ flops.

In order to enhance the accuracy of the method, one may increase M while N is kept fixed at a moderate value. Thus, the cost of the algorithm may be rather high. In Section 5, we show by some numerical experiments that a better strategy is to increase N instead, while M is fixed at a moderate value.

3.2. Numerical experiments. Throughout this section, we consider the model integral $I_k^{[0, 1]}(f, g)$ with

$$g(x) = \sqrt{x^2 + 3x + 4},$$

and some k and f . It can be seen that $g'(x) > 0$, $x \in [0, 1]$. Then we apply Algorithm I and illustrate the error estimate (30).

Experiment 1. Here we study the case when k and N are kept fixed and $h \rightarrow 0$. According to the error bound (30), the convergence of the method is of order $\mathcal{O}(h^N)$, which can be illustrated by the following example.

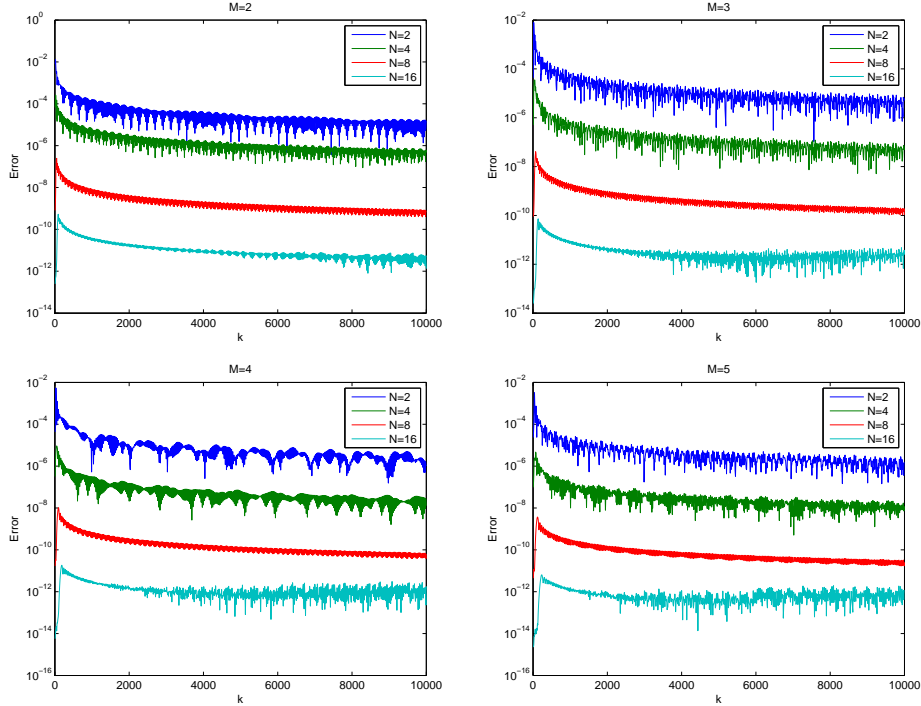
Let $k = 100$, and

$$f(x) = \frac{x^{4.5}}{1 + x^2}.$$

Since $f \in C^4[0, 1]$ and $g \in C^\infty[0, 1]$, $F \in C^4[g(0), g(1)]$. Thus, the error bound (30) is valid if the parameter N does not exceed 3. The absolute errors with their rates, as $h \rightarrow 0$, are given in Table 1, for $N = 1, 2$, and 3.

TABLE 1. Numerical results of Experiment 1

M	$N = 1$		$N = 2$		$N = 3$	
	error	rate	error	rate	error	rate
16	$3.9820E-5$		$1.4024E-7$		$1.4064E-9$	
32	$3.6917E-6$	3.4	$2.3716E-9$	5.9	$1.6626E-11$	6.4
64	$8.2527E-7$	2.2	$1.2537E-10$	4.2	$7.4073E-13$	4.5
128	$2.0128E-7$	2.0	$7.5501E-12$	4.1	$4.2413E-14$	4.1
256	$5.0018E-8$	2.0	$4.6807E-13$	4.0	$2.4046E-15$	4.1
512	$1.2486E-8$	2.0	$2.9164E-14$	4.0	$3.1165E-17$	6.3

FIGURE 1. The error scaled by k of Algorithm I with some M and N , when applied to the model integral of Experiment 2.

As mentioned in §3.1, the distribution of the interpolating nodes in each panel tends to the Gaussian one if h is small enough. Thus, the convergence rate will be higher than what we expect. For coarser grids (corresponding to larger h), instead, the convergence rate of $\mathcal{O}(h^{N+1})$ is not expected. Considering all these facts together, the numerical results of Table 1 are in agreement with the theory. Experiment 2. The error bound (30) suggests that, for a fixed parameters N and M , the error of the method decays with the rate $\mathcal{O}(k^{-1})$ as $k \rightarrow \infty$. In this experiment we illustrate this result by a numerical example.

Take $f(x)$ the same as in Experiment 1. For each $M \in \{2, 3, 4, 5\}$ and $N \in \{2, 4, 8, 16\}$, we apply Algorithm I to the model integral $I_k^{[0,1]}(f, g)$ when k varies in a wide band from 10 to 10,000. The absolute error is scaled by k , and the scaled error for each N and M has been illustrated as a function of k by Figure 1.

As it is seen the error for each M and N does not deteriorate as k increases, and this observation is in agreement with the convergence order of $\mathcal{O}(k^{-1})$, suggested by the theory.

4. NO STATIONARY POINTS: METHOD II

In this section, we follow [4] and employ an efficient interpolation method which does not allow the Runge effect. In addition, one can always increase its accuracy to as high as desired without increasing the computational cost.

4.1. Algorithm II. Consider the integral (1) on an arbitrary interval $[a, b]$ (not necessarily of small length). Here, we do not divide the integration interval $[a, b]$ into smaller subintervals as in Algorithm I. Instead, we consider the N -point MFCC rule on the whole interval and approximate the values $\tilde{F}(t_{j,N})$ by an accurate interpolating process as follows.

For a large integer N' , consider a set of points $-1 = d_0 < \dots < d_{N'} = 1$, as defined in §2.1. Choose a positive integer $s < N'$ and fix it throughout the process. For each $x \in [-1, 1]$, select an s -tube $\mathcal{N}(x) = (d_j, \dots, d_{j+s-1})$ such that $d_j \leq x \leq d_{j+s-1}$. Such a selection is not necessarily unique, while it is always possible since $s < N'$. In [4], some selecting strategies are introduced, and in [15] a MATLAB code of a certain selecting strategy has been provided. Then define $\tilde{Q}'_N \varphi(x)$ by the Lagrange polynomial interpolation of $\varphi(x)$ at nodes of $\mathcal{N}(x)$, i.e.

$$\tilde{Q}'_N \varphi(x) = \ell(x) \sum_{n=j}^{j+s-1} \frac{w_n}{x - d_n} \varphi(d_n), \quad (31)$$

where

$$\ell(x) = \prod_{n=j}^{j+s-1} (x - d_n), \quad w_n = \frac{1}{\prod_{m \neq n} (d_n - d_m)}.$$

The formula (31) is called by Rutishauser [25] as the “first form of the barycentric interpolation formula”. The formula is one which has been mentioned earlier in §3.1. In [11] it has been proved that the formula is backward stable.

An error bound for the method can be obtained as follows. If $f \in C^s[a, b]$ and $g \in C^{s+1}[a, b]$, $F \in C^s[a, b]$, and then

$$\max_{0 \leq n \leq N} |(\tilde{F} - \tilde{Q}_N \tilde{F})(t_{n,N})| \leq C' \lambda^s,$$

where

$$\lambda = \max_{1 \leq n \leq N'} |d_n - d_{n-1}|,$$

and C' is a generic constant independent of N and N' , but dependent of s . Thus, by Theorem 2.1, the total error of integration over $[a, b]$ is bounded by

$$C k^{-r} (b - a)^{m+1-r} N^{-m+\rho(r)} + C' k^{-1} \lambda^s \sqrt{N}, \quad (32)$$

where C is a generic constant independent of N , N' , and s .

On the other hand, one can easily see from (27) that λ decreases with the rate $\mathcal{O}(N'^{-1})$ as $N' \rightarrow \infty$. Therefore, if we take $N' \geq kN$, the error bound (32) is reduced to

$$C k^{-r} (b - a)^{m+1-r} N^{-m+\rho(r)} + C' k^{-s-1} N^{-s+1/2}. \quad (33)$$

The discussion about the complexity of the method is the same as that of Method I. Since the total interpolation process requires $\mathcal{O}(Ns^2)$ flops, the whole algorithm is performed at the cost of $\mathcal{O}(N(\ln N + s^2))$. In order to increase the accuracy of the interpolation, one needs to increase N' only, while the parameter s is usually kept fixed at a moderate value. Thus, the cost of computation does not grow. Also, the N -point FCC rule converges rapidly as N increases, so by a moderate value of N , one can reach a rather high accuracy. Therefore, the integral (1) can be accurately approximated by Algorithm II at a rather low cost. We have carried out a set of numerical experiments in Section 5 to illustrate our claim here.

TABLE 2. Numerical results of Experiment 1

	$s = 2$		$s = 3$		$s = 4$		$s = 5$	
N	error	rate	error	rate	error	rate	error	rate
2	6.4200E-4		6.4200E-4		6.4200E-4		6.4200E-4	
4	5.4239E-4	0.2	5.4239E-4	0.2	5.4239E-4	0.2	5.4239E-4	0.2
8	9.9097E-5	2.5	9.9075E-5	2.5	9.9075E-5	2.5	9.9075E-5	2.5
16	2.9253E-6	5.1	2.9263E-6	5.1	2.9263E-6	5.1	2.9263E-6	5.1
32	1.4805E-8	7.6	1.7694E-9	10.7	1.7329E-9	10.1	1.7330E-9	10.7
64	1.2081E-9	3.6	1.7068E-12	10.0	2.2446E-15	19.6	1.1572E-15	20.5

4.2. Numerical experiments. Throughout this section, we consider the model integral $I_k^{[-1,1]}(f, g)$ with

$$f(x) = \frac{x-1}{1+x^2}, \quad g(x) = \sqrt{x^2 + 3x + 4},$$

and some k . It is easy to see that $g'(x) > 0$, $x \in [-1, 1]$, so Algorithm II can be applied. The aim of this subsection is to illustrate the convergence estimate (33). Experiment 1. If k is kept fixed, the error bound (33) decays with the rate $\mathcal{O}(N^{-s+1/2})$ as N increases. We consider the assumed model integral with $k = 100$, and apply the algorithm with $s \in \{2, 3, 4, 5\}$, some increasing N , and $N' = kN$. The absolute errors with their rates are gathered in Table (2). As it is seen, the experimental convergence rate for each s is in agreement with the theoretical one.

Experiment 2. If N , N' , and s are kept fixed, the error bound (33) convergence with the rate $\mathcal{O}(k^{-2})$ as $k \rightarrow \infty$. In this set of experiments, we consider the assumed model integral with k varying in a wide band from 10 to 10,000. Algorithm II with $s \in \{2, 3, 4, 5\}$ and some values of N is applied while $N' = kN$. The absolute error is scaled by k^2 and the scaled error for each s and N has been illustrated as a function of k by Figure 2.

The horizontal trend of the scaled errors illustrates the theoretical convergence rate $\mathcal{O}(k^{-2})$.

5. COMPARISONS

In this section, we compare Methods I and II to see which one in practice achieves a given accuracy faster. Our answer to this question, is based on some numerical results accompanied by a rough theoretical discussion.

Recall from Section 3 that Algorithm I requires $\mathcal{O}(MN(N^2 + \ln N))$ flops. From the error bound (30), one can see that a rather large N may reduce the error effectively even if $h < 1$ is not too small. This is because N is appearing in the exponents. Also, it is not recommended to use very large N , since the complexity of the algorithm increases rapidly. Thus, N is better to take moderate values while M should be rather small.

Similarly, we recall from Section 4 that the number of required flops for Algorithm II is $\mathcal{O}(N(\ln N + s^2))$. Note that the exponents in the error bound (33) may not be as large as the exponents in the error bound (30) because s usually takes rather small values in practice; larger values increases the complexity of the algorithm rapidly.

Based on the discussion above, our guess is that Algorithm I may be faster than Algorithm II provided that M is rather small. In the following, we carry out a set of numerical experiments to support our guess. All the computations in this section are performed on a computer with 4.0 GB RAM memory and 3.0 GHz dual core processor.

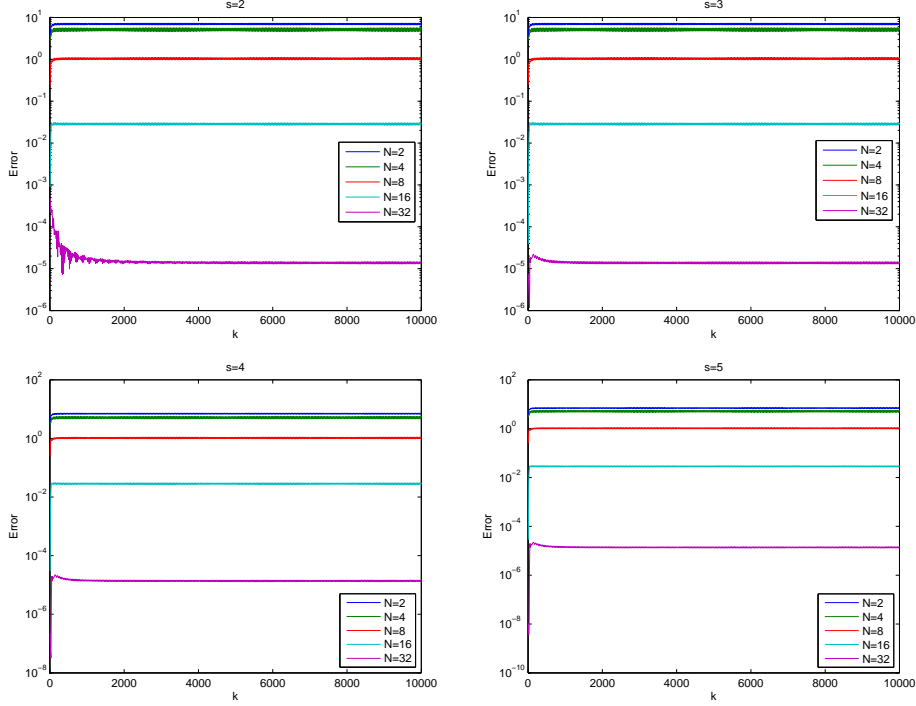


FIGURE 2. The error scaled by k of Algorithm II with some s and N , when applied to the model integral of Experiment 2.

Recalling the model integral in §4.2 with $k = 1000$, consider the integral

$$\int_{-1}^1 \frac{x-1}{1+x^2} \exp\left(1000i\sqrt{x^2+3x+4}\right) dx. \quad (34)$$

In Algorithm I, M is kept fixed, and the accuracy increases to the desired one by letting $N \rightarrow \infty$ (see Figure 3). The algorithm for each set of the parameters $\{M, N\}$ is implemented 100 times and the average of CPU times (in seconds) versus the relative error is plotted by an asterisk (*).

In algorithm II, s is kept fixed and the accuracy increases by letting $N \rightarrow \infty$ while $N' = kN$ (see Figure 3). The algorithm for each set of the parameters $\{s, N\}$ is implemented 100 times and the average of CPU times (in seconds) versus the relative error is plotted by a square (\square).

Figure 3 consists of eight subplots, the first four correspond with smaller values of M , and the others correspond with larger values of M . In each subplot, the range of N for one algorithm may differ from that for the other one since it is determined such that a given accuracy is achieved by that algorithm. For example, in the first subplot, corresponding with $(M, s) = (2, 2)$, the range of N for Algorithm I is 2(2)20, while for Algorithm II it is 4(4)48. As it is seen, Algorithm I is more accurate only if M is not too large.

For further support of the above conjecture, we carry out a different experiment on the same sample integral (34). In Algorithm I, N is fixed by 5 and the accuracy increases by letting M grow as $M = 2^m$, $m = 1, \dots, 10$ (see Figure 4). The algorithm for each set of the parameters $\{M, N\}$ is implemented 10 times and the average of CPU times (in seconds) versus the relative error is plotted by an asterisk (*). Since performance of the algorithm for larger M is rather time consuming, we have repeated each experiment only 10 times.

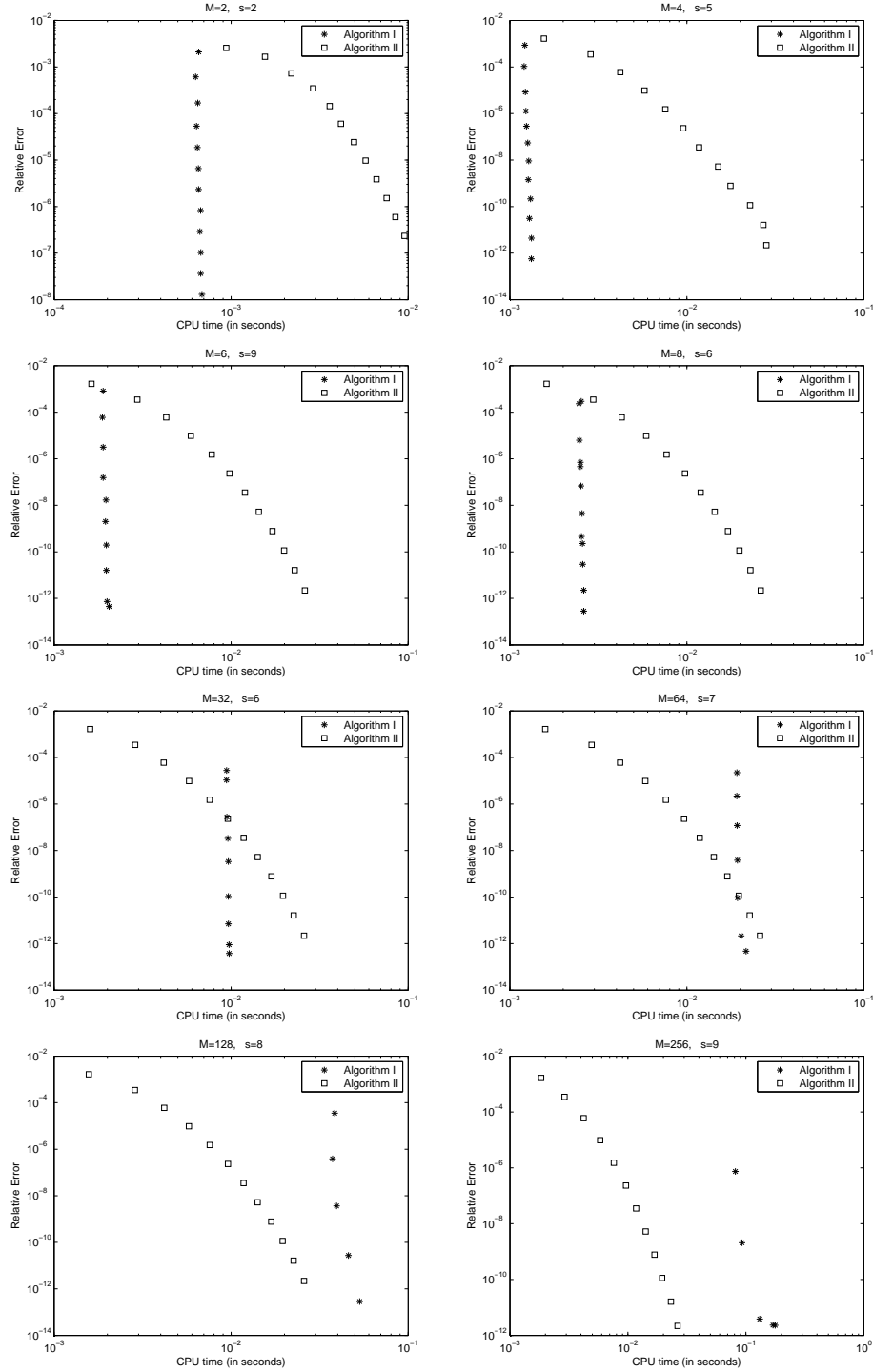


FIGURE 3. Average CPU time (in seconds) vs. relative error of Algorithms I and II when applied on the model integral (34). The accuracy in each algorithm increases by growing N .

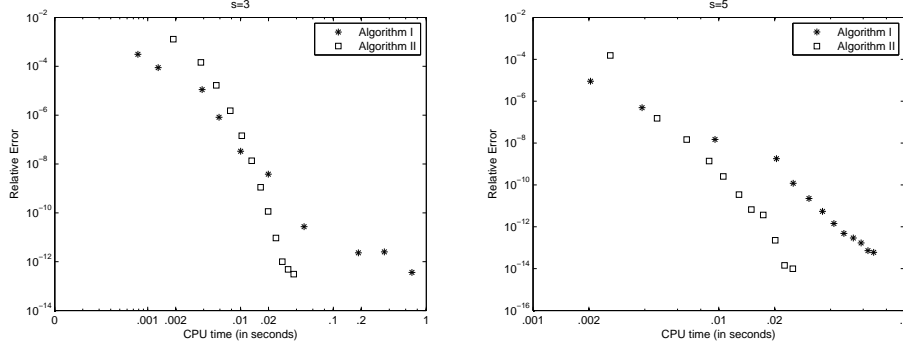


FIGURE 4. Average CPU time (in seconds) vs. relative error of Algorithms I and II when applied on the model integrals (34) (left) and (35) (right). In Algorithm I, $N = 5$, and the accuracy increases by growing M .

In algorithm II, s is fixed by 3 and the accuracy increases by letting N grow as $5(5)60$ while $N' = kN$ (see Figure 4, left). The algorithm for each set of the parameters $\{s, N\}$ is implemented 100 times and the average of CPU times (in seconds) versus the relative error is plotted by a square (\square).

Only when $M \leq 32$ (corresponding to the first five asterisks), Algorithm I is more accurate.

We have repeated the experiment for the model integral of §3.2 with $k = 100$, i.e.

$$\int_0^1 \frac{x^{4.5}}{1+x^2} \exp\left(100i\sqrt{x^2+3x+4}\right) dx. \quad (35)$$

In comparison to the previous experiment corresponding to Figures 3 and 4 (left), the values of the parameters in each algorithm may differ, but the main strategy for choosing them is the same. The results, which are similar to the previous experiment for the integral (34), have been depicted by Figures 5 and 4 (right). Note that in this experiment the model integral (35) has an endpoint singularity at 0.

6. IN THE PRESENCE OF STATIONARY POINTS

Consider the integral (1), and assume that $g'(x)$ vanishes at a finite number of points in $[a, b]$. It is said that a function $\varphi(x)$ has a stationary point of order $n \geq 1$ at $\xi \in [a, b]$ if

$$\varphi'(\xi) = \dots = \varphi^{(n)}(\xi) = 0, \quad \varphi^{(n+1)}(\xi) \neq 0.$$

If $[a, b]$ is divided at the stationary points of g and some other points between them, then g has no stationary points in each subinterval except at one of its endpoints. Thus, without loss of generality, we consider the integral $I_k^{[a,b]}(f, g)$, where g has a single stationary point of order n at a , and $g'(x) > 0$ for any $x \in (a, b]$.

Under the change of variable $\tau = g(x)$, the integral is reduced to $I_k^{[g(a), g(b)]}(F)$, with F defined by (9). Again by the change of variable $\tau = g(a) + l\hat{x}$ with $l = g(b) - g(a)$, the integral is transformed to the standard interval $[0, 1]$:

$$I_k^{[g(a), g(b)]}(F) = l \exp(ikg(a)) I_{lk}^{[0,1]}(\hat{F}), \quad (36)$$

where

$$\hat{F}(\hat{x}) = F(g(a) + l\hat{x}). \quad (37)$$

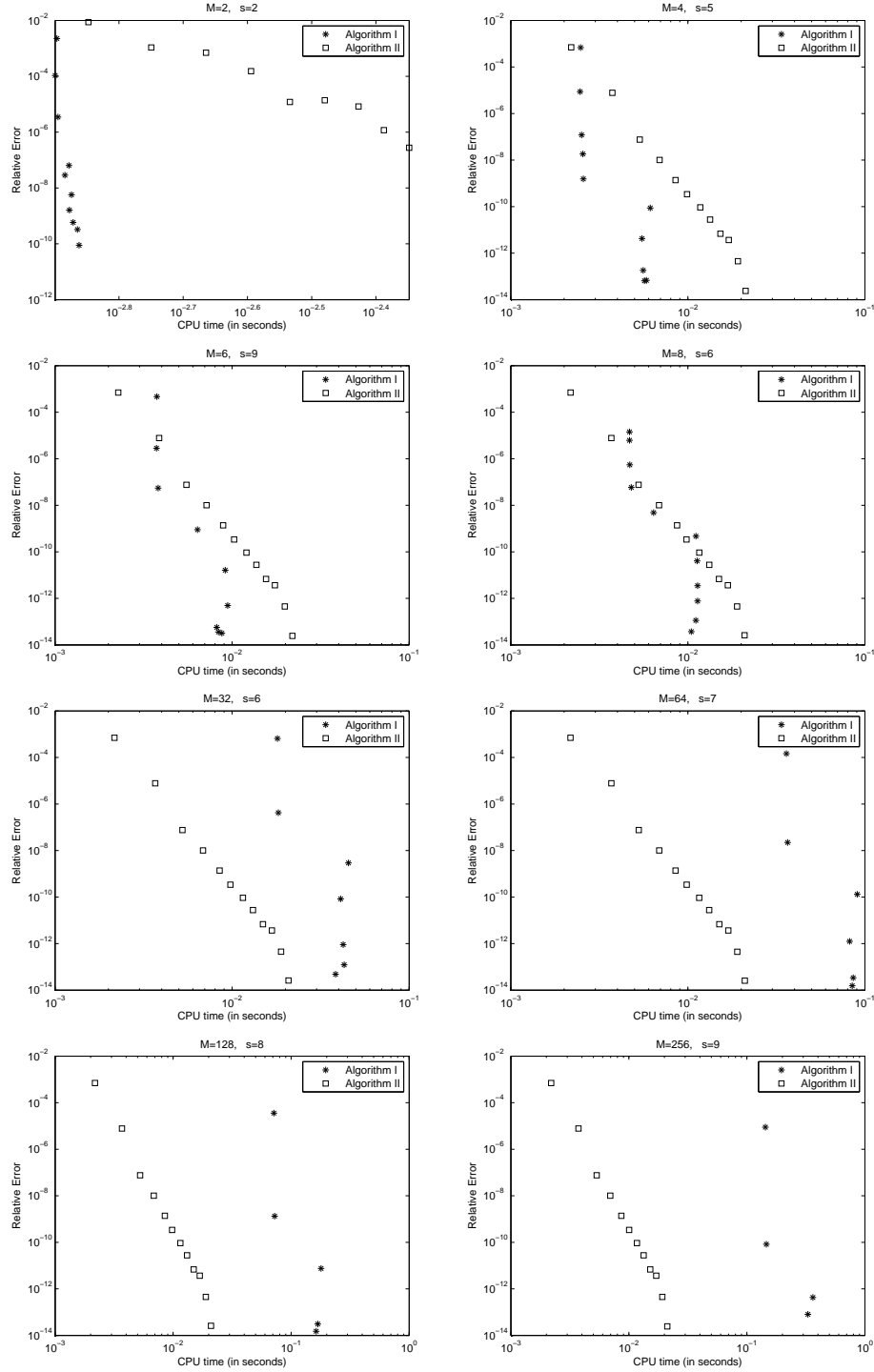


FIGURE 5. Average CPU time (in seconds) vs. relative error of Algorithms I and II when applied on the model integral (35). The accuracy in each algorithm increases by growing N .

By Theorem 4.1 of [8], if f and g are smooth enough, $\widehat{F} \in C_\beta^m[0, 1]$ for $\beta = -n/(n+1)$ and some m , depending on the degrees of smoothness of f and g . Recall from [8] that for any $\beta < 0$, $C_\beta^m[0, 1]$ is defined as the space of all $\varphi \in C(0, 1]$ such that

$$\|\varphi\|_{m,\beta} := \max \left\{ \sup_{x \in (0,1]} |x^{j-\beta} \varphi^{(j)}(x)|, \quad j = 0, \dots, m \right\} < \infty. \quad (38)$$

Now the composite algorithm, introduced in [8], can be applied for computing $I_{lk}^{[0,1]}(\widehat{F})$. The algorithm employs the classical graded partition

$$\Pi_{M,q} := \left\{ x_j := \left(\frac{j}{M} \right)^q : j = 0, 1, \dots, M \right\} \quad (39)$$

of the integration interval $[0, 1]$, for some grading exponent $q > 1$ sufficiently large. Then in each panel $[x_{j-1}, x_j]$, the N -point FCC rule is applied.

Our proposed algorithm here is almost the same. The only difference is that we employ the N -point MFCC rule in each panel. In the following, we show how the N -point MFCC rule may be employed, and how it affects the total error of the algorithm.

6.1. The composite MFCC rules on graded meshes. The method introduced here, is to use the N -point MFCC rule (16) with $N' = N$ and $u_j = t_{j,N}$, i.e. N Clenshaw-Curtis points. As it is seen, the only difference of the algorithm here from Algorithm I, lies in the type of the partitioning $0 = x_0 < \dots < x_M = 1$: Uniform meshes for Algorithm I and graded meshes for the current algorithm. The following theorem provides an error bound for the algorithm, proposed in this section.

Theorem 6.1. *Let f and g be so smooth that $\widehat{F} \in C_\beta^{N+1}[0, 1]$, and let $g(x)$ has a single stationary point at $x = a$ of order $n \geq 1$. Let $0 \leq r < 1 + \beta$ with $\beta = -n/(n+1)$ and choose*

$$q > (N+1-r)/(\beta+1-r). \quad (40)$$

Then an error bound for the composite MFCC rule on the graded mesh (39) is

$$Ck^{-r}M^{-N-1+r}\|\widehat{F}\|_{N+1,\beta} + C'k^{-1}\sqrt{N}M^{-N} \quad (41)$$

where C and C' are constants independent of k and M .

Proof. Theorem 2.1 implies that the total error of a composite MFCC rule, is the sum of the error of the composite FCC rule and the sum over all the panels of the interpolating errors multiplied by $C'k^{-1}\sqrt{N}$.

As mentioned earlier, $\widehat{F} \in C_\beta^{N+1}[0, 1]$ if f and g are smooth enough. Thus, the first term, i.e. the error of the composite FCC rule, is estimated by Theorem 3.6 of [8] as

$$Ck^{-r}M^{-N-1+r}\|\widehat{F}\|_{N+1,\beta}. \quad (42)$$

For the second term of the error, the results in [24] or [2, §8.3] imply that the maximum interpolating error in each panel is of order $\mathcal{O}(M^{-N-1})$. Note that, since $\beta < 0$, the integration over the first panel $[x_0, x_1]$ is simply approximated by 0 (see Eq. (3.5) in [8]), so the interpolating error in the first panel vanishes. Thus, the second term of the error estimate can be written as

$$C'k^{-1}\sqrt{N}(M-1)M^{-N-1}, \quad (43)$$

and this with (42) yields the result. \square

TABLE 3. Numerical results of Experiment 1

M	$N = 4$		$N = 6$		$N = 8$	
	error	rate	error	rate	error	rate
100	$3.2120E-5$		$5.9080E-6$		$4.8773E-6$	
200	$1.2603E-7$	8.0	$4.7397E-8$	7.0	$7.8279E-9$	9.3
400	$9.4778E-9$	3.7	$1.6276E-10$	8.2	$1.7216E-11$	8.8
800	$2.1893E-10$	5.4	$2.3113E-12$	6.1	$2.6714E-14$	9.3

6.2. Numerical experiments. In this set of experiences we try to illustrate the error estimate (41) by some numerical results. Consider the following sample integral

$$\int_0^1 \frac{x-1}{1+x^2} \exp(ikx^4) dx, \quad (44)$$

with some $k > 0$. Clearly 0 is the only stationary point of the oscillator function, and it is of order 3. Also, the amplitude and the oscillator functions are so smooth that $\hat{F} \in C_\beta^{N+1}[0, 1]$ for any $N > 0$. Thus the composite MFCC rules can be applied.

Experiment 1. If M increases while the other parameters are kept fixed, Theorem 6.1 implies that the convergence rate is of order $\mathcal{O}(M^{-N})$. In order to illustrate this result, consider the sample integral (44) with $k = 1000$. We employed the N -point composite MFCC rule on the graded mesh (39) with $q = \lfloor (N+1)/(\beta+1) \rfloor + 1$ for some N and M . For some M and N , the absolute errors $E_{M,N}$ are given in Table 3. Also the rates of convergence, as M increases and N is kept fixed, are given. The rate, corresponding to each pair (M, N) , is computed by the following formula:

$$\text{rate} = \log_2 \frac{E_{M/2,N}}{E_{M,N}}.$$

It is seen that the error decays as $\mathcal{O}(M^{-N})$.

Experiment 2. The error estimate (41) implies that the order of convergence is $\mathcal{O}(k^{-1})$ if $k \rightarrow \infty$ and other parameters remain fixed. In order to illustrate this result, we again consider the sample integral (44) with k varying in a wide band from 10 to 10,000. For $M = 200$ and $N \in \{4, 6, 8\}$, we apply the composite MFCC rule on the graded mesh (39) with $q = \lfloor (N+1)/(\beta+1) \rfloor + 1$. The absolute error is scaled by k , and the scaled error for each s and N has been illustrated as a function of k by Figure 6.

7. CONCLUSIONS

We have introduced a general method, which can generate various modifications of the FCC rules and their composite versions. The modified algorithms can be applied to the oscillatory integral (1), while they do not deal with the inverse of g . The main tool that allows us to design such modifications is the interpolation; any kind of the modifications corresponds with a certain interpolation method.

When g has no stationary points in $[a, b]$, two kinds of modified FCC rules have been introduced. For each kind, an error estimate has been given theoretically, and then illustrated by some numerical experiments. Also, some numerical experiments have been carried out for a comparison of the accuracy and the efficiency of the two rules.

When g has a finite number of stationary points in $[a, b]$, the idea has been applied to the composite quadrature rules, introduced in [8]. Similar to the previous case, an error estimate has been given theoretically, and then illustrated by some numerical experiments.

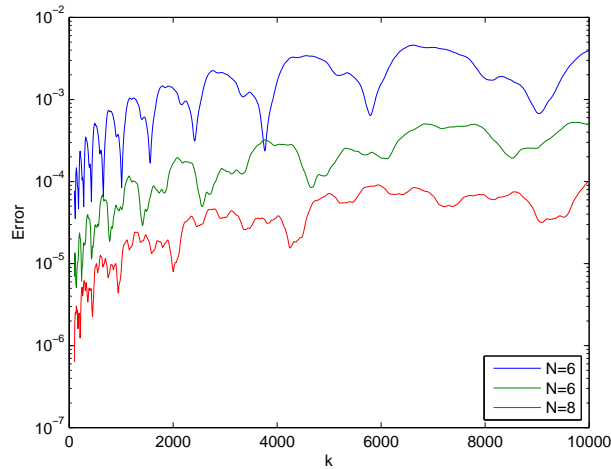


FIGURE 6. Absolute error as a function of k of the composite MFCC rule on the graded mesh (39); the involved parameters are $M = 200$, $N \in \{4, 6, 8\}$, and $q = \lfloor (N + 1)/(\beta + 1) \rfloor + 1$.

REFERENCES

- [1] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications, New York, 1965.
- [2] K.E. Atkinson. *The numerical solution of integral equations of the second kind*. Cambridge University Press, Cambridge, 1997.
- [3] Jean-Paul Berrut and L.N. Trefethen. Barycentric Lagrange interpolation. *SIAM Rev.*, 46(3):501–517, 2004.
- [4] O. Bruno and D. Hoch. Numerical differentiation of approximated functions with limited order-of-accuracy deterioration. *SIAM J. Numer. Anal.*, 50(3):1581–1603, 2012.
- [5] S.N. Chandler-Wilde, I.G. Graham, S. Langdon, and E.A. Spence. Numerical-asymptotic boundary integral methods in high-frequency scattering. *Acta Numerica*, 21:89–305, 2012.
- [6] C.W. Clenshaw and A.R. Curtis. A method for numerical integration on an automatic computer. *Numer. Math.*, 2(1):197–205, 1960.
- [7] Alfredo Deaño and Daan Huybrechs. Complex Gaussian quadrature of oscillatory integrals. *Numer. Math.*, 112:197–219, 2009.
- [8] V. Domínguez, I.G. Graham, and T. Kim. Filon-clenshaw-curtis rules for highly oscillatory integrals with algebraic singularities and stationary points. *SIAM J. Numer. Anal.*, 51(3):1542–1566, 2013.
- [9] V. Domínguez, I.G. Graham, and V.P. Smyshlyaev. Stability and error estimates for Filon-Clenshaw-Curtis rules for highly oscillatory integrals. *IMA J. Numer. Anal.*, 31:1253–1280, 2011.
- [10] L.N.G. Filon. On a quadrature formula for trigonometric integrals. *Proc. Roy. Soc. Edinburgh*, 49:38–47, 1928.
- [11] N.J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA J. Numer. Anal.*, 24(4):547–556, 2004.
- [12] Daan Huybrechs and Sheehan Olver. Superinterpolation in highly oscillatory quadrature. *Found. Comput. Math.*, 12:203–228, 2012.
- [13] Daan Huybrechs and Stefan Vandewalle. On the evaluation of highly oscillatory integrals by analytic continuation. *SIAM J. Numer. Anal.*, 44:1026–1048, 2006.
- [14] A. Iserles and S.P. Nørsett. Efficient quadrature of highly oscillatory integrals using derivatives. *Proc. R. Soc. Lond. A*, 461:1383–1399, 2005.
- [15] Hassan Majidian. Creating stable quadrature points with preassigned points by interpolation. to appear.
- [16] Joaquim R.R.A. Martins, P. Sturdza, and J.J. Alonso. The complex-step derivative approximation. *ACM Trans. Math. Softw.*, 29(3):245–262, 2003.
- [17] Sheehan Olver. Moment-free numerical approximation of highly oscillatory integrals with stationary points. *Euro. J. Appl. Maths.*, 18:435–447, 2007.

- [18] Sheehan Olver. *Numerical Approximation Of Highly Oscillatory Integrals*. PhD thesis, University of Cambridge, 2008.
- [19] R. Piessens. Computing integral transforms and solving integral equations using Chebyshev polynomial approximations. *J. Comput. Appl. Math.*, 121:113–124, 2000.
- [20] R. Piessens and M. Branders. The evaluation and application of some modified moments. *BIT*, 13:443–450, 1973.
- [21] R. Piessens and M. Branders. Numerical solution of integral equations of mathematical physics, using Chebyshev polynomials. *J. Comput. Phys.*, 21:178–196, 1976.
- [22] R. Piessens and M. Branders. Computation of Fourier transform integrals using Chebyshev series expansions. *Computing*, 32:177–186, 1984.
- [23] R. Piessens and F. Poleunis. A numerical method for the integration of oscillatory functions. *BIT*, 11:317–327, 1971.
- [24] J.R. Rice. On the degree of convergence of nonlinear spline approximation. In I. J. Schoenberg, editor, *Approximations with special emphasis on spline functions*, pages 349–365. 1969.
- [25] Heinz Rutishauser. *Vorlesungen über numerische Mathematik*, volume 1. Birkhäuser, Basel, Stuttgart, 1976; English translation, *Lectures on Numerical Mathematics*, Walter Gautschi, ed. Birkhäuser, Boston, 1990.
- [26] I.H. Sloan and W.E. Smith. Product integration with the Clenshaw-Curtis and related points. Convergence properties. *Numer. Math.*, 30:415–428, 1978.
- [27] I.H. Sloan and W.E. Smith. Product integration with the Clenshaw-Curtis points: implementation and error estimates. *Numer. Math.*, 34:387–401, 1980.
- [28] Shuhuang Xiang. Efficient Filon-type methods for $\int_a^b f(x)e^{i\omega g(x)} dx$. *Numer. Math.*, 105:633–658, 2007.

DEPARTMENT OF BASIC SCIENCES, IRANIAN INSTITUTE FOR ENCYCLOPEDIA RESEARCH, PO BOX 14655-478, TEHRAN, IRAN.

E-mail address: majidian@iecf.ir